```
1
2
3   from a0_items import *
4
5
6   import psutil
7   import datetime
8
9   ##########################################################################################
    ##########
10  ### plugins - Check hardware specs
11  ##########################################################################################
    ############
12
13  #
    https://www.juniper.net/documentation/us/en/software/junos/automation-scripting/topics/ta
    sk/junos-python-modules-psutil-module.html
14  ### *** CPU FUNCTIONS ***
15
16  def CPU_count ():
17
18      count_CPU = psutil.cpu_count()
19      # Number of logical CPUs in the system
20      print ("psutil.cpu_count() = {0}".format(count_CPU))
21
22      return count_CPU
23
24  #CPU_count ()
25
26
27
28  ### *** DISK FUNCTIONS ***
29
30  def DISK_partitions ():
31
32      # List of named tuples containing all mounted disk partitions
33      dparts = psutil.disk_partitions()
34      print("psutil.disk_partitions() = {0}".format(dparts))
35
36      return dparts
37
38
39  def DISK_usage ():
40
41      # Disk usage statistics
42      du = psutil.disk_usage('/')
43      print("psutil.disk_usage('/') = {0}".format(du))
44
45      return du
46
47
48  ### *** MEMORY FUNCTIONS ***
49
50  def get_sys_memory ():
51
52  # System memory usage statistics
53      mem = psutil.virtual_memory()
54      print("psutil.virtual_memory() = {0}".format(mem))
55
56      THRESHOLD = 100 * 1024 * 1024  # 100MB
57      if mem.available <= THRESHOLD:
58          print("warning, available memory below threshold")
59
60      return mem
61
62
63  #get_sys_memory ()
```

```
64
65
66
67
68  ### *** PROCESS FUNCTIONS ***
69
70  def get_process_id ():
71
72      # List of current running process IDs.
73      pids = psutil.pids()
74      print("psutil.pids() = {0}".format(pids))
75
76      return pids
77
78  #get_process_id ()
79
80
81
82
83  def get_process_id_name ():
84
85      # Check whether the given PID exists in the current process list.
86      for proc in psutil.process_iter():
87
88          try:
89
90              pinfo = proc.as_dict(attrs=['pid', 'name'])
91
92          except psutil.NoSuchProcess:
93
94              pass
95
96          else:
97
98            print(pinfo)
99
100
101 ### *** SYSTEM INFORMATION FUNCTIONS ***
102
103 def get_sys_boot_time ():
104
105     # System boot time expressed in seconds since the epoch
106     boot_time = psutil.boot_time()
107     print("psutil.boot_time() = {0}".format(boot_time))
108
109     # System boot time converted to human readable format
110     print(datetime.datetime.fromtimestamp(psutil.boot_time()).strftime("%Y-%m-%d %H:%M:%S"
        ))
111
112     return boot_time
113
114
115
116 def get_sys_users ():
117
118     # Users currently connected on the system
119     users = psutil.users()
120     print("psutil.users() = {0}".format(users))
121
122     return users
123
124 #get_sys_users ()
```